# ISC3N

ENCODER INTERFACE WITH DIGITAL I/O

PCI Version

# USER'S MANUAL

## INTRODUCTION

The ISC3N card is a PCI bus compatible Encoder Interface with Digital Inputs and Outputs that provides a direct transfer of position values from incremental encoders to the PC without any external position readout units. The additional digital input/output lines make the ISC3N card ideal for use in motion control systems, measurement applications, position regulations, etc. It is possible to connect up to three incremental encoders, and 16 digital inputs and outputs to the PC with a single ISC3N card.

## FEATURES

- Counting in positive and negative direction
- Trigger initiated counting
- Trigger latched counting
- Position latched counting
- Reset/Preset
- Counting error detection (phase error)
- 16 programmable I/O lines divided in four groups
- 3 output lines may be programmed to output equality status bits
- 3 inputs as External triggers or additional inputs
- Interrupt request may be generated by Trigger, Error Detection or Position Equality
- Fast industry standard PCI interface

## ISC3N CHARACTERISTICS

- Three 32-bit counters
- Input signals from incremental encoders:
    1. A, B, RI +5V TTL compatible
    2. A+, A-, B+, B-, RI+, RI- RS422 compatible
- Digital input filters 2 stages
- Input pulse width, low or high - $t_W$ 60 ns min. (see figure below)
- Phase difference between any A and B signal transitions - $t_{AB}$ 30 ns min. (see figure below)
- Digital Inputs and Outputs +5V TTL compatible, buffered, 47kΩ pull-ups
- Output current 25 mA max.
- Power consumption 200 mA max.
- IRQ line INTA#
- PCI 5V, 32-bit, 33MHz compatible, PCI 3.3V optional

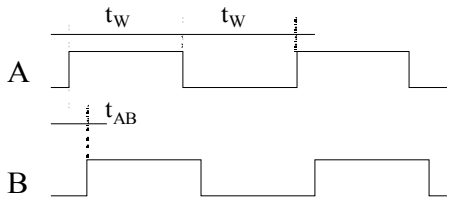



Figure 1: Encoder input signal requirements

## DESCRIPTION

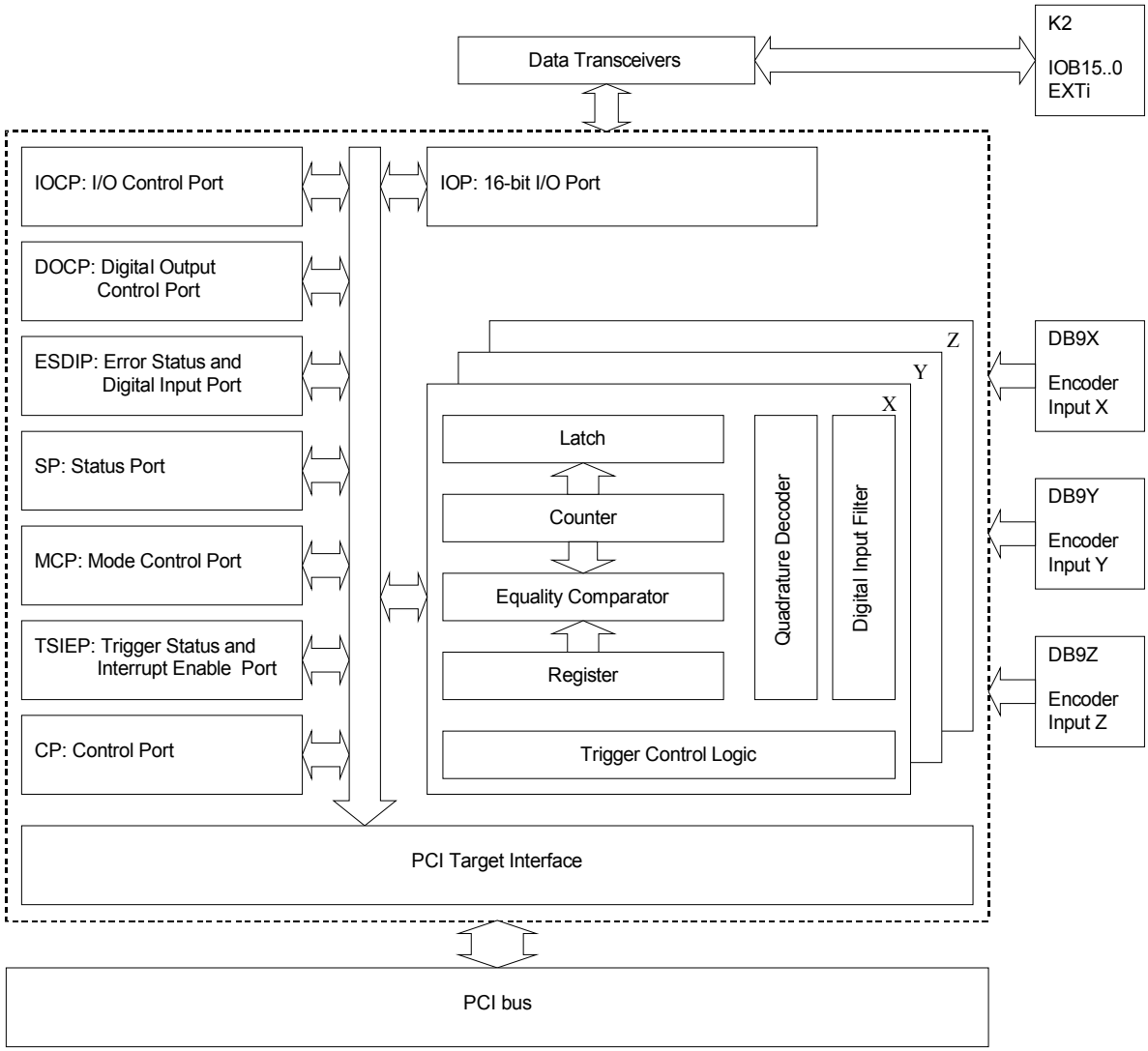
Each encoder input channel of the ISC3N consists of a line receiver, quadrature decoder with phase error detection, control logic for trigger evaluation, 32-bit counter, 32-bit latch, 32-bit register and 32-bit equality comparator. It is possible to reset and preset the counter. The data from the counter is accessed only through the latch. The ISC3N can be programmed to operate in several different modes:

- counter counts independently of any trigger signals,
- counter starts counting when the trigger signal is detected,
- latch follows the counter, that is, the latch is transparent,
- counter data is latched whenever the trigger signal is detected,
- all counters are latched whenever selected counter data equals to register data (latching at desired position).

## INSTALLATION

Power off your PC and insert the ISC3N card in a free PCI expansion slot. Upon power-up the operating system should detect the card. Then simply follow the Driver Installation Wizard.

## BLOCK DIAGRAM



## CONNECTOR PIN ASSIGNMENT

- **DB9X, Y, Z** Encoder Connectors, Sub-D DB9 female

| +5V TTL | RS422 | Pin No.: | | RS422 | +5V TTL |
|---|---|---|---|---|---|
| GND | GND | **1** | **6** | RI– | |
| RI | RI+ | **2** | **7** | B– | |
| B | B+ | **3** | **8** | A– | |
| A | A+ | **4** | **9** | GND | GND |
| +5V | +5V | **5** | | | |

- **K2** I/O Connector, HD26 male or Sub-D DB25 female

| HD26 | Pin No.: | | HD26 |
|---|---|---|---|
| GND | **1** | **2** | EXTX+ |
| EXTX– | **3** | **4** | EXTY+ |
| EXTY– | **5** | **6** | EXTZ+ |
| EXTZ– | **7** | **8** | +5V |
| GND | **9** | **10** | IOB0 |
| IOB1 | **11** | **12** | IOB2 |
| IOB3 | **13** | **14** | IOB4 |
| IOB5 | **15** | **16** | IOB6 |
| IOB7 | **17** | **18** | IOB8 |
| IOB9 | **19** | **20** | IOB10 |
| IOB11 | **21** | **22** | IOB12 |
| IOB13 | **23** | **24** | IOB14 |
| IOB15 | **25** | **26** | GND |

| DB25 | Pin No.: | | DB25 |
|---|---|---|---|
| GND | **1** | **14** | EXTX+ |
| EXTX– | **2** | **15** | EXTY+ |
| EXTY– | **3** | **16** | EXTZ+ |
| EXTZ– | **4** | **17** | +5V |
| GND | **5** | **18** | IOB0 |
| IOB1 | **6** | **19** | IOB2 |
| IOB3 | **7** | **20** | IOB4 |
| IOB5 | **8** | **21** | IOB6 |
| IOB7 | **9** | **22** | IOB8 |
| IOB9 | **10** | **23** | IOB10 |
| IOB11 | **11** | **24** | IOB12 |
| IOB13 | **12** | **25** | IOB14 |
| IOB15 | **13** | | |

The EXT*i* inputs are normally used as external triggers, but may also be used as additional general-purpose digital inputs.
The IOB0..15 pins are programmable I/O lines divided in four groups by four. Each group may be programmed to function as inputs or as outputs.
The pins IOB0..2 may be individually programmed to output the Count=Register (C=R*i*) equality status of the channels X, Y and Z, respectively.

Total loading of the +5V pins should be kept below 200mA. A short-circuit from these pins to any other pin must be avoided.

Connectors **K3** and **K4** are reserved for factory test.

# ISC3N Registers

| BAR | +3 | +2 | +1 | +0 |
|---|---|---|---|---|
| +00 | XCNT - X Counter | | | |
| +04 | XREG - X Register | | | |
| +08 | YCNT - Y Counter | | | |
| +0C | YREG - Y Register | | | |
| +10 | ZCNT - Z Counter | | | |
| +14 | ZREG - Z Register | | | |
| +18 | | | IOP | |
| +1C | | | | IOCP |
| +20 | SP | MCP | TSIEP | CP |
| +24 | | | DOCP | ESDIP |

### XCNT, YCNT, ZCNT – X, Y, Z Encoder Counters / offsets 00, 08, 10h

| 0 | 31 | 0 |
|---|---|---|
| | /CNT | |

### XREG, YREG, ZREG – X, Y, Z Compare Registers / offsets 04, 0C, 14h

| 04 | 31 | 0 |
|---|---|---|
| | /REG | |

### IOP – I/O Port / offset 18h

| 18 | 31 | 16 | 15 | 0 |
|---|---|---|---|---|
| | | | IOB[15:0] | |

### IOCP – I/O Control Port / offset 1Ch

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | GDD3 | GDD2 | GDD1 | GDD0 |

| GDD*i* | Group Data Direction |
|---|---|
| 0 | Inputs |
| 1 | Outputs |

Bit GDD0 controls IOB[3:0], GDD1 controls IOB[7:4], GDD2 controls IOB[11:8] , GDD3 controls IOB[15:12].

### CP - Control Port / offset 20h

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | CI | DL | | | | |

| CI | Count Inhibit |
|---|---|
| 0 | Counters enabled |
| 1 | Counters stopped |

| DL | Counter Read Latch Control |
|---|---|
| 0 | Counter data is latched |
| 1 | Read Latch is transparent and follows counter data |

### TSIEP - Trigger Select and Interrupt Enable Port / offset 21h

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | ERIE | EQIE | TRIE | | TSZ | TSY | TSX |

| TS*i* | Trigger Source |
|---|---|
| 0 | Reference Impulse (RI*i*) |
| 1 | External Trigger (IN*i*) |

| ERIE | Error Interrupt Enable |
|---|---|
| 0 | Error interrupt disabled |
| 1 | Enable interrupt on quadrature decoder error (ERR*i*) |

| EQIE | Equality Interrupt Enable |
|---|---|
| 0 | Equality interrupt disabled |
| 1 | Enable interrupt on equality (C=R*i*) |

| TRIE | Trigger Interrupt Enable |
|---|---|
| 0 | Trigger interrupt disabled |
| 1 | Enable trigger interrupt (T*i*) |

**MCP - Mode Control Port / offset 22h**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | ZM1 | ZM0 | YM1 | YM0 | XM1 | XM0 |

| *i*M1 | *i*M0 | Counter Mode |
|---|---|---|
| 0 | 0 | Continuous counting mode |
| 0 | 1 | Latch counter data on trigger (T*i*) |
| 1 | 0 | Start counter after trigger (T*i*) |
| 1 | 1 | Latch all counters on equality (C=R*i*) |

The *i*M1 M0 bits control the count enable (*i*M0) and latch function (*i*M1) of the selected channel, except when *i*M0 M1=11, they also affect other two counters.

**SP - Status Port / offset 23h**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | TZ | TY | TX | | C=RZ | C=RY | C=RX |

**T*i*** Logic High indicates that the corresponding trigger signal has been detected. Bits are cleared by writing zeros to them.

**C=R*i*** Logic High indicates that the counter data equalled the corresponding register. Bits are cleared by writing zeros to them.

**ESDIP - Error Status and Digital Input Port / offset 24h**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | ERRZ | ERRY | ERRX | | INZ | INY | INX |

**ERR*i*** Logic High indicates that the built-in quadrature decoder detected an invalid transition of the encoder AB input signals. Bits are cleared by writing zeros to them.

**IN*i*** These bits are read-only and their values correspond to the logic level of the respective K4 connector input pins EXT*i*.

**DOCP - Digital Output Control Port / offset 25h**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | OCZ | OCY | OCX | | | | |

| **OC*i*** | **Digital Output Control** |
|---|---|
| 0 | Connector pins IOB[2:0] correspond to bits in I/O PORT |
| 1 | Connector pins IOB[2:0] correspond to C=RZ, C=RY, C=RX status bits in SP, respectively. |

This feature enables routing equality status to the outside world via IOB[2:0] port signals.

# SOFTWARE REFERENCE

## EIPCI32.DLL functions description

---

The **EIPCILibOpen** opens and allocates the resources used by the driver. This function must be matched with a call to EIPCILibClose.

Int EIPCILibOpen();

Parameters:
none

Result of
0: OK
2: Driver can not be opened.
NOTE:
The EIPCILibOpen function has to be called first to open the driver.

---

The **EIPCILibClose** closes and releases the resources used by the driver.

VOID EIPCILibClose();

Parameters:
none

NOTE:
This function must be matched with a call to EIPCILibOpen.

---

The ISC3N initalization routine. The **EIPCIInit** function:

Detects the installed card,
resets I/O Port and I/O Control Port,
resets Trigger Select and Interrupt Enable Port, resets Mode Control Port, resets Status Port,
resets Error Status and Digital Input Port,
resets Digital Output Control Port,
resets X, Y and Z axis position.

int EIPCIInit ();

Parameters:
none

Result of
0: OK
1: Card not installed
2: Driver not opened.
NOTE:
The EIPCIInit function detects the card and initializes the cards parameters. It is usually called immediatelly after the EIPCILibOpen call.

---

The ISC3N initalization routine. The **EIPCIInitEx** function:

Detects the installed card,
resets I/O Port and I/O Control Port,
resets Trigger Select and Interrupt Enable Port, resets Mode Control Port, resets Status Port,
resets Error Status and Digital Input Port,
resets Digital Output Control Port.

int EIPCIInitEx ();

Parameters:
none

Result of
0: OK
1: Card not installed
2: Driver not opened.
NOTE:
The same as EIPCIInit except it doesn't reset the X, Y and Z counters.

---

The **EIPCIReadAxis** function returns the Counter Value (current position) of the selected axis, accessed through the Latch as a signed 32 bit value.

```
int EIPCIReadAxis (
  BYTE Axis,                // Axis selected
  BYTE DLMode               // Data Latch mode
);
```

Parameters:

Axis

One of the three channels of the ISC3N Card selected

1 : X Axis
2 : Y Axis
3 : Z Axis

DLMode

DLMode defines the DL bit of the ISC3N Control Port while reading position

0 : DL bit of Control Port is set to 1 while reading position data from the Latch. After reading the data the DL bit remains set to 1.
1 : DL bit of Control Port is set to 0 while reading position data from the Latch. After reading the data the DL bit is set to 1.
2 : DL bit of Control Port is set to 0 while reading position data from the Latch. After reading the data the DL bit remains set to 0.

---

The **EIPCIPresetAxis** function presets the Counter of the selected axis with signed 32 bit Value.

```
VOID EIPCIPresetAxis (
  BYTE Axis,                // Axis selected
  int  Value                // Preset Value
);
```

Parameters:

Axis

One of the three channels of the ISC3N Card selected

1 : X Axis
2 : Y Axis
3 : Z Axis

Value

Preset Value

---

The **EIPCIReadRegister** function returns the axis Register Value as a signed 32 bit value.

```
int EIPCIReadRegister (
  BYTE Axis,                    // Axis selected
);
```

Parameters:
  Axis
    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis

---

The **EIPCIPresetRegister** function writes signed 32 bit Value to the axis Register. Corresponding C=R bit of the Status Port is set when the Counter data equals the Register data.

```
VOID EIPCIPresetRegister (
  BYTE Axis,                    // Axis selected
  int  Value                    // Preset Value
);
```

Parameters:
  Axis
    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis
  Value
    Preset Value

---

The **EIPCIResetAxis** function presets the axis Counter with value of 0.

```
VOID EIPCIResetAxis (
  BYTE Axis                     // Axis selected
);
```

Parameters:
  Axis    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis

---

The **EIPCISetCountingMode** function sets the axis counting mode.

```
VOID EIPCISetCountingMode (
  BYTE Axis,                  // Axis selected
  BYTE CountingMode       // Counting mode selected
);
```

Parameters:
  Axis
    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis
  CountingMode
    CountingMode byte defines the counting mode of the selected axis
      0 : Continuous Counting Mode
      1 : Trigger Latched Counting Mode
      2 : Trigger Started Counting Mode
      3 : Equality Latched Counting Mode

---

The **EIPCIGetEqualityStatus** function returns C=R (D0-D2) bits of the Status Port.

```
BYTE EIPCIGetEqualityStatus ();
```

Parameters:
  none

---

The **EIPCIResetEqualityStatus** function resets C=R (D0-D2) bits of the Status Port.

```
VOID EIPCIResetEqualityStatus ();
```

Parameters:
  none

---

The **EIPCIGetTriggerStatus** function returns T (D4-D6) bits of the Status Port.

```
BYTE EIPCIGetTriggerStatus ();
```

Parameters:
  none

---

The **EIPCIResetTriggerStatus** function resets T (D4-D6) bits of the Status Port.

```
VOID EIPCIResetTriggerStatus (
  BYTE Axis                    // Axis selected
);
```

Parameters:
  Axis
    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis

---

The **EIPCISetTriggerSource** function selects Trigger Source.

```
VOID EIPCISetTriggerSource (
  BYTE Axis,                          // Axis selected
  BOOL ExternalTrigger      // External trigger flag
);
```

Parameters:
  Axis
    One of the three channels of the ISC3N Card selected
      1 : X Axis
      2 : Y Axis
      3 : Z Axis
  ExternalTrigger
    ExternalTrigger flag defines Trigger Source of the selected axis
      FALSE : the Reference Impulse as Trigger Source
      TRUE  : the External Trigger as Trigger Source

---

The **EIPCIGetErrorStatus** function returns ERR (D4-D6) bits of the Error Status and Digital Input Port.

```
BYTE EIPCIGetErrorStatus ();
```

Parameters:
  none

---

The **EIPCIResetErrorStatus** function resets ERR (D4-D6) bits of the Error Status and Digital Input Port.

```
VOID EIPCIResetErrorStatus ();
```

Parameters:
  none

---

The **EIPCISetIOP** function writes the 16 bit Value to the I/O Port.

```
VOID EIPCISetIOP (
  WORD Value                // Value
);
```

Parameters:
  Value
    Byte value, written to the I/O Port

---

The **EIPCISetIOCP** function writes the Value to the I/O Control Port.

```
VOID EIPCISetIOCP (
  BYTE Value                        // Value
);
```

Parameters:
  Value
    Byte value, written to the I/O Control Port

---

The **EIPCISetCP** function writes the Value to the Control Port.

```
VOID EIPCISetCP (
  BYTE Value                        // Value
);
```

Parameters:
  Value
    Byte value, written to the Control Port

---

The **EIPCISetTSIEP** function writes the Value to the Trigger Select and Interrupt Enable Port.

```
VOID EIPCISetTSIEP (
  BYTE Value                    // Value
);
```

Parameters:
  Value
    Byte value, written to the Trigger Select and Interrupt Enable Port

---

The **EIPCISetMCP** function writes the Value to the Mode Control Port.

```
VOID EIPCISetMCP (
  BYTE Value                    // Value
);
```

Parameters:
  Value
    Byte value, written to the Mode Control Port

---

The **EIPCISetSP** function writes the Value to the Status Port.

```
VOID EIPCISetSP (
  BYTE Value                    // Value
);
```

Parameters:
  Value
    Byte value, written to the Status Port

---

The **EIPCISetESDIP** function writes the Value to the Error Status and Digital Input Port.

```
VOID EIPCISetESDIP (
  BYTE Value                    // Value
);
```

Parameters:
  Value
    Byte value, written to the Error Status and Digital Input Port.
Note:
  Bits D0 to D3 of the ESDIP are read-only.

---

The **EIPCISetDOCP** function writes the Value to the Digital Output Control Port.

```
VOID EIPCISetDOCP (
  BYTE Value                    // Value
);
```

Parameters:
  Value
    Byte value, written to the Status Port

---

The **EIPCIGetIOP** function returns D0-D15 bits of the I/O Port.

```
WORD EIPCIGetIOP ();
```

Parameters:  none

---

The **EIPCIGetIOCP** function returns D0-D7 bits of the I/O Control Port.

```
BYTE EIPCIGetIOCP ();
```

Parameters:
  none

---

The **EIPCIGetCP** function returns D0-D7 bits of the Control Port.

```
BYTE EIPCIGetCP ();
```

Parameters:
  none

---

The **EIPCIGetTSIEP** function returns D0-D7 bits of the Trigger Select and Interrupt Enable Port.

```
BYTE EIPCIGetTSIEP ();
```

Parameters:
  none

---

The **EIPCIGetMCP** function returns D0-D7 bits of the Mode Control Port.

BYTE EIPCIGetMCP ();

Parameters:
none

---

The **EIPCIGetSP** function returns D0-D7 bits of the Status Port.

BYTE EIPCIGetSP ();

Parameters:
none

---

The **EIPCIGetESDIP** function returns D0-D7 bits of the Error Status and Digital Input Port.

BYTE EIPCIGetESDIP ();

Parameters:
none

---

The **EIPCIGetDOCP** function returns D0-D7 bits of the Digital Output Control Port.

BYTE EIPCIGetDOCP ();

Parameters:
none